



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/628,959	07/28/2003	Eitan Hefetz	34874-020 UTIL	6174
64280	7590	05/13/2009	EXAMINER	
MINTZ, LEVIN, COHN, FERRIS, GLOVSKY & POPEO, P.C.			PATEL, MANGLESH M	
ONE FINANCIAL CENTER				
BOSTON, MA 02111			ART UNIT	PAPER NUMBER
			2178	
			MAIL DATE	DELIVERY MODE
			05/13/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/628,959	Applicant(s) HEFETZ ET AL.
	Examiner MANGLESH M. PATEL	Art Unit 2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If no period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 26 February 2009.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-9, 18-20 and 26 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-9, 18-20 and 26 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/1449)
 Paper No(s)/Mail Date _____

4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date _____

5) Notice of Informal Patent Application
 6) Other: _____

DETAILED ACTION

1. This Non-Final action is responsive to the RCE filed on 2/26/2009.

2. In the continuation Claims 1-9, 18-20 and 26 are pending. Claims 10-17 and 21-25 have been canceled. Claims 1, 6, 18 and 26 are the independent claims.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

4. Claims 1-9, 18-20 and 26 remain rejected under 35 U.S.C. 102(e) as being anticipated by Dulepet (U.S. 7,316,003, filed Dec 18, 2002).

Regarding **Independent claim 1**, a computer-implemented method comprising:

- Providing a design-time translator and a run-time translator that both correspond to a defined page element, The run-time translator and design time translator configured on a processor;

- During design-time for a page, invoking the design-time translator for a page template including the defined page element having content components, said design-time invoking resulting in the defined page element in the page template being translated into a design-time representation of the content components in the page , the design-time representation being rendered in accordance with a predefined layout of a container for the content components, the page template being available to a plurality of remote users of a portal, the content components including a first content component and a second content component, the first content component configured as static content with a run-time behavior determinable at design-time, and the second component configured as dynamic content with a run-time behavior not determinable at run-time, such that at design-time a tag is used to represent the dynamic content on the page rendered at design-time; and

- During run-time for the page, invoking the run-time translator for the page template, said run-time invoking resulting in the content components being obtained and the defined page element in the

page template being translated into a run-time presentation of the obtained one or more content components in accordance with the layout of the container, wherein the second component configured as dynamic content is determined and obtained in parallel, at run-time using threading, with other dynamic content stored in blocks without ordering in a content storage medium to render the dynamic content of the second component rather than the tag used during design-time.

Dulepet teaches creation/editing of a dynamic web page using a WYSIWYG editor. He describes the use of a design time engine which in response to a controller-deployed dynamic page request, the design time engine replaces the dynamic code JSP elements with a design time component, such a component comprises a content placeholder representative of content that would have been generated by a JSP container if the controller had deployed the dynamic code element to an executing JSP container. Furthermore the page template is available to a plurality of remote users of a portal because the Merged model in fig 2 synchronizes the updated from the editor in design-time to the application database server thus making it available to the remote users of the portal. He then describes that during run-time upon receiving the source code, JSP container replaces dynamic source code elements within the source code with dynamically generated page content, and returns a dynamically generated web page (see abstract, fig 2-3, column 1, lines 5-67, column 2, lines 10-50, column 2, lines 55-67, column 3, lines 1-52, column 6, lines 5-58 & column 5, lines 50-67). Furthermore the dynamic component is determined and obtained in parallel at run-time using threading because Dulepet discloses that controller 230 may deploy web page source code/or Java servlets to multiple JSP containers. Dulepet (see column 6, lines 55-67) provides a suggestion that controller 230 handles 2 or more concurrently running tasks for processing multiple JSP containers thereby resulting in rendering of the dynamic content components.

Regarding Dependent claim 2, which depends on claim 1, Dulepet discloses wherein said invoking the design-time translator further results in presentation of a WYSIWYG layout editor using the design-time representation of the one or more content components in the page (column 2, lines 10-50, column 2, lines 55-67, including the explanation provided in the Independent claims).

Regarding Dependent claim 3, which depends on claim 2, Dulepet discloses wherein the said invoking the design-time translator further results in client-side scripting components being included in the design-time representation to form at least part of the WYSIWYG layout editor and enable adding a content component to a content container using a

drag-and-drop action (column 2, lines 10-50, column 2, lines 55-67, including the explanation provided in the Independent claims).

Regarding Dependent claim 4, which depends on claim 2, Dulepet discloses wherein the page template comprises a portal page template, and the WYSIWYG layout editor comprises a WYSIWYG portal page layout editor (column 6, lines 5-58 & column 5, lines 50-67, including the explanation provided in the Independent claims).

Regarding Dependent claim 5, which depends on claim 4, Dulepet discloses wherein the defined page element comprises a custom Java Server Page tag and the design-time translator and the run-time translator comprise Java Server Page tag handlers for the custom Java Server Page tag, and wherein the run-time translator obtains portal dynamic content according to the portal page template and the design-time translator does not (column 1, lines 5-67, column 2, lines 10-50, column 2, lines 55-67, column 3, lines 1-52, including the explanation provided in the Independent claims).

Regarding Independent claims 6 and 18, an article comprising a machine-readable storage medium storing instructions operable to cause one or more machines to perform operations comprising: Providing a design-time translator and a run-time translator that both correspond to a defined page element, the run-time translator and design time translator configured on a processor; during design-time for a page, invoking the design-time translator for a page template including the defined page element having content components, said design-time invoking resulting in the defined page element in the page template being translated into a design-time representation of the content components in the page, the design-time representation being rendered in accordance with a predefined layout of a container for the content components, the page template being available to a plurality of remote users of a portal, the content components including a first content component and a second content component, the first content component configured as static content with a run-time behavior determinable at design-time, and the second component configured as dynamic content with a run-time behavior not determinable at run-time, such that at design-time a tag is used to represent the dynamic content on the page rendered at design-time; and during run-time for the page, invoking the run-time translator for the page template, said run-time invoking resulting in the content components being obtained and the defined page element in the page template being translated into a run-time presentation of the obtained one or more content components in accordance with the layout of the container, wherein the second component configured as dynamic content is determined and obtained in parallel, at run-time using threading, with other dynamic content stored

in blocks without ordering in a content storage medium to render the dynamic content of the second component rather than the tag used during design-time.

Dulepet teaches creation/editing of a dynamic web page using a WYSIWYG editor. He describes the use of a design time engine which in response to a controller-deployed dynamic page request, the design time engine replaces the dynamic code JSP elements with a design time component, such a component comprises a content placeholder representative of content that would have been generated by a JSP container if the controller had deployed the dynamic code element to an executing JSP container. Furthermore the page template is available to a plurality of remote users of a portal because the Merged model in fig 2 synchronizes the updated from the editor in design-time to the application database server thus making it available to the remote users of the portal. He then describes that during run-time upon receiving the source code, JSP container replaces dynamic source code elements within the source code with dynamically generated page content, and returns a dynamically generated web page (see abstract, fig 2-3, column 1, lines 5-67, column 2, lines 10-50, column 2, lines 55-67, column 3, lines 1-52, column 6, lines 5-58 & column 5, lines 50-67). Furthermore the dynamic component is determined and obtained in parallel at run-time using threading because Dulepet discloses that controller 230 may deploy web page source code/or Java servlets to multiple JSP containers. Dulepet (see column 6, lines 55-67) provides a suggestion that controller 230 handles 2 or more concurrently running tasks for processing multiple JSP containers thereby resulting in rendering of the dynamic content components.

Regarding Dependent claim 7, which depends on claim 6, Dulepet discloses wherein translating the placeholder during design-time comprises adding code enabling editing of the portal page, the added code forming at least part of the WYSIWYG portal layout editor (column 2, lines 10-50, column 2, lines 55-67, including the explanation provided in the Independent claims).

Regarding Dependent claim 8, which depends on claim 7, Dulepet discloses wherein the added code comprises client-side scripting that enables addition of a content component to a content container in the portal page using a drag-and-drop action (column 2, lines 10-50, column 2, lines 55-67, including the explanation provided in the Independent claims).

Regarding Dependent claim 9, which depends on claim 6, Dulepet discloses wherein the placeholder comprises a custom Java Server Page tag, said translating the placeholder during design-time comprises invoking a design-time Java Server Page tag handler corresponding to the custom Java Server Page tag, and said translating the placeholder during run-time comprises invoking a run-time Java Server Page tag handler corresponding to the custom Java Server Page tag (column 1, lines 5-67, column 2, lines 10-50, column 2, lines 55-67, column 3, lines 1-52, including the explanation provided in the Independent claims).

Regarding Dependent claim 19, which depends on claim 18, Dulepet discloses wherein the first tag handler interprets the portal page template by including client-side scripting that enables addition of a content component to a content container in the portal page template using a drag-and-drop action (column 2, lines 10-50, column 2, lines 55-67, including the explanation provided in the Independent claims)

Regarding Dependent claim 20, which depends on claim 18, the claim describes a system that contains the same limitations as claim 5 and is rejected under the same rationale.

Regarding Independent claim 26, Dulepet discloses A computer-implemented method for selectively interpreting a portal page layout template, the method comprising: providing a design-time translator and a run-time translator, the design-time translator and the run-time translator both corresponding to a same defined page element or placeholder, and being invoked based on a current mode of operation, the run-time translator and design time translator configured on a processor (See fig 4a & 4B & 4C & abstract showing a design time translator and run-time translator corresponding to a defined page element and being invoked by the pre/post processors); translating a placeholder in a portal template during design-time into a representation of a container designed to present portal content using a single template file for both run-time and design-time, the container representation showing a layout context for the portal content that will be obtained and revealed at run-time, the container representation also directly presenting dynamic content source information for the content container (See fig 4a & 4B & 4C & abstract & column 6, lines 25-64, which shows translating a placeholder during design-time in 4A into a representation of a container using a template file as disclosed in both 4A and 4B.);presenting a WYSIWYG portal layout editor using the container representation designed to present the portal dynamic content, the WYSIWYG portal layout editor facilitating editing of the portal template and the resulting portal page (column 6, lines 1-59, wherein the WYSIWYG editor is presented using the container

representation for editing of the portal template and portal page); obtaining portal dynamic content during run-time from a dynamic content source, the placeholder in the portal template being translated into a presentation of the container containing the obtained portal dynamic content component, wherein the dynamic content is determined and obtained in parallel, at run-time using threading with other dynamic content stored in blocks without ordering in a dynamic content source providing a storage medium, to render the dynamic content rather than a tag used during design-time to represent the dynamic content (see fig 4a-4c, showing dynamic content during run-time); and parsing and locating, by a run-time application, the placeholders in the template and replacing them with the run-time content components, and at design-time, parsing and rendering the same template with a representation of the content components, in place of the actual run-time content components, to reveal the run-time layout during design of the template (see fig 4a-4c & column 6, lines 1-67, disclosing placeholders in the template being replaced with the run-time content, including design-time previews of the dynamic content with placeholders). **Furthermore the dynamic component is determined and obtained in parallel at run-time using threading because Dulepet discloses that controller 230 may deploy web page source code/or java servlets to multiple JSP containers. Dulepet (see column 6, lines 55-67) provides a suggestion that controller 230 handles 2 or more concurrently running tasks for processing multiple JSP containers thereby resulting in rendering of the dynamic content components.**

It is noted that any citation [[s]] to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. [[See, MPEP 2123]]

Response to Arguments

5. Applicant's arguments filed 2/26/2009 have been fully considered but are not persuasive.

Applicant Argues: Dulepet uses "dynamic Source code" to determine "dynamically generated web page content" rather than using a simple "tag" (pg 12, paragraph 2)

The Examiner Respectfully Disagrees: Dynamic content is obtained using JSP (see column 5, lines 20-32) which comprises tags, JSP contains its own tag library for performing custom actions.

Applicant Argues: Dulepet is completely silent with respect to any obtaining dynamic content at run-time in parallel using threading and staging that the dynamic content in a storage medium before rendering the dynamic content at run-time (pg 12, paragraph 4 – pg 13, paragraph 1)

The Examiner Respectfully Disagrees: The dynamic component is determined and obtained in parallel at run-time using threading because Dulepet discloses that controller 230 may deploy web page source code/or java servlets to multiple JSP containers. Dulepet (see column 6, lines 55-67) provides a suggestion that controller 230 handles 2 or more concurrently running tasks for processing multiple JSP containers thereby resulting in rendering of all the dynamic content components at run-time.

(Note: The Examiner appreciates applicant's effort to expedite prosecution in this application. However as explained above the claim limitations are not clear in its current state to overcome the reference. The Examiner advises applicant to contact the examiner for an interview to discuss potential novel language within the specification and clarify over the teachings of the prior art to expedite prosecution).

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Manglesh M. Patel whose telephone number is (571) 272-5937. The examiner can normally be reached on M, W 6 am-3 pm T, TH 6 am-2pm, Fr 9am-6pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen S. Hong can be reached on (571) 272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Application/Control Number: 10/628,959
Art Unit: 2178

Page 9

/Manglesh M Patel/

Manglesh Patel
Examiner, Art Unit 2178
May 8, 2009

	/CESAR B PAULA/ Primary Examiner, Art Unit 2178
--	--